Network Simulation

Minerva CS166
Anna Pauxberger
25 March 2018

Modifications

Prevent social splits

Small communities may want to prevent complete social splits between two members in their community, by encouraging them to talk to each other. That can either lead to their opinions becoming closer (update_1_1) or their relationship to improve (update_1_2). There are 2 modifications that each have 2 variations. The first modification is preventing nodes to split. Instead of breaking apart when the weight is less than 0.05, opinions or relationships are edited. The first variation (update_1_1) decreases the difference of opinions, by introducing another parameter epsilon, which makes alpha as large as possible and therefore brings opinions together faster. It has a lower bound of 0, and an upper bound of 0.5/alpha. The second variation (update_1_2) sets the weight of the edge back to 0.2.

Multiple topics

People tend to have more than one opinion. The model shows simulations in which people are allowed to have 2 (update_2_1) or 3 (update_2_2) opinions, and shows the subsequent clustering in 4 and 8 groups via a vector input. For subsequent calculations, the average difference of opinions is taken into account.

An interesting addition could be to not take the average of the opinions but to allow a split if the opinions are far apart. Averaging eventually will lead to convergence since opinions can only move towards each other, but allowing nodes to split or worsen their relationship when only one opinion is bad could have interesting clustering behavior.

The model assumes a near normal distribution for the Watts-Strogatz graph, and a power-law distribution for the Barabasi-Albert graph. The parameters alpha, beta, and gamma are set to the defaults 0.03, 0.3 and 4 respectively. Alpha is bounded by 0 and 0.5, and brings opinions closer together faster as it increases. Beta is bounded by 0 and 1, and determines how fast the weight of an edge changes. Gamma is bounded by 0 and (1+1/beta). If gamma is smaller or equal to 1, edge weights will converge to 1.

The update function for opinions and weights, respectively, are as follows:

$$\Delta o_i = \alpha\, w_{ij}\, (o_j - o_i)$$

$$\Delta w_{ij} = \beta\, w_{ij}\, (1 - w_{ij})\, (1 - \gamma\, |o_i - o_j|\,)$$

Local Analysis

Mathematical Analysis for epsilon

The only parameter that was relevant to be bounded was the epsilon parameter in update_1_1.

delta o = epsilon * alpha * weight * (opinion i - opinion j)

The amplifier epsilon should increase alpha to as high as possible, since it should bring the opinions as close together as possible. Alpha has to be a value between 0 and 0.5, thus the new value epsilon*alpha has to be bounded by 0 and 0.5 as well.

e*a < 0.05
e < 0.5/a
e < 0.5/0.03
0 < e <16.67

Given the inequality that epsilon*alpha has to be smaller than 0.05, we can solve for epsilon having to be smaller than 0.5/alpha. Different alpha values will allow different values for epsilon. The given alpha = 0.03 defines the upper bound of epsilon to be 16.67. The lower bound of 0 remains.

Vector Field Plot Analysis for Split Prevention

As seen in Table 1, the original update (update_1_0) allows for nodes to lose their connection when the difference in opinion is very high, and eventually when the weight of the connection is below 0.05. This is indicated via the purple lines, and the arrows pointing towards the 0.05 line. Update_1_1 shows the effect of bringing opinions closer. Instead of purple lines, the green lines indicate that once the 0.05 boundary has been crossed, the difference in opinions decreases (pointing left to a smaller opinion difference). Update_1_2 shows the effect of strengthening ties. Since ties cannot break, all lines are green and bounce up to 0.2 whenever a relationship reaches a weight of below 0.05.

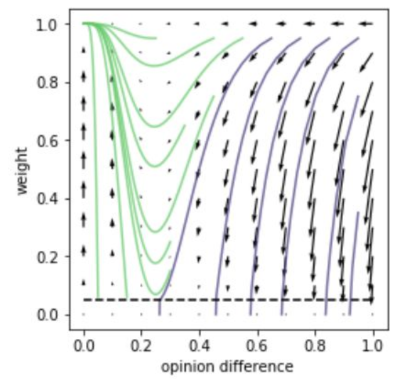
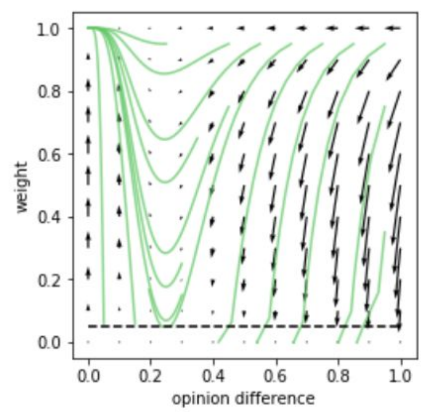
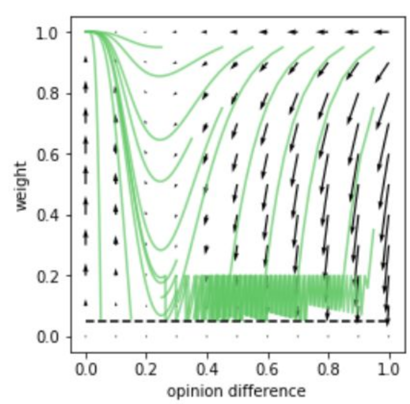
update	code	graph
update_1_0	new_weight = 0	
update_1_1	new_opinion = ow[-1][0] + 10 * delta_o	
update_1_2	new_weight = 0.2	

Table 1: Vector field plots

Implementation

First, I ran the simulations on a Watts-Strogatz graph, which initializes the graph with random connections based on a near normal distribution. However, to more accurately simulate social dynamics, I decided to go with the Barabasi-Albert graph instead, which relies on preferential attachment that chooses strongly connected nodes more frequently than weakly connected nodes. This creates a scale free network following the power law, since the choice of a connecting node is not independent of the number of neighbors anymore. A few nodes a very highly connected, while a large number have only few connections. However, the Barabasi-Albert graph showed the exact same densities and clusterings for update_1_1 and update_1_2, which is why I was intrigued to compare the two graphs with each other.

Simulation overview

To keep track of the several simulations that include different parameters,
the code is structured as follows:

SocialDynamicsSimulation	original
SocialDynamicsSimulation2	inherits original, avoids splits
SocialDynamicsSimulation3	edited for 2 topics (opinions)
SocialDynamicsSimulation4	edited for 3 topics (opinions)

Table 1: Simulation Overview

Updates Overview

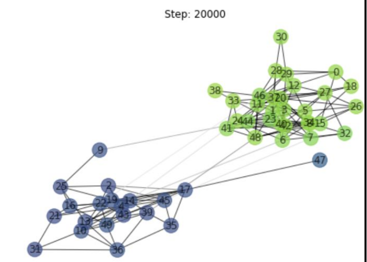
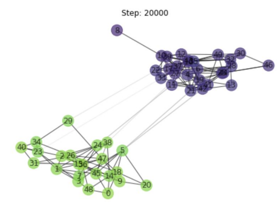
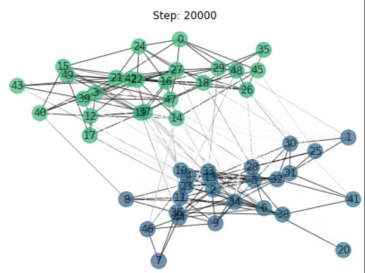
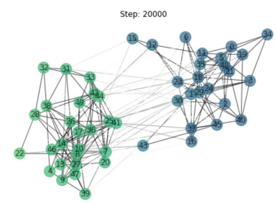
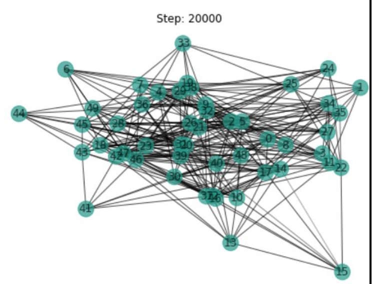
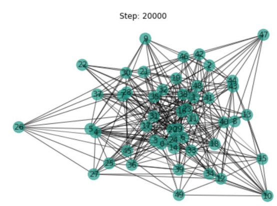
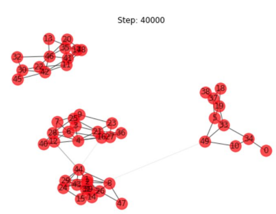
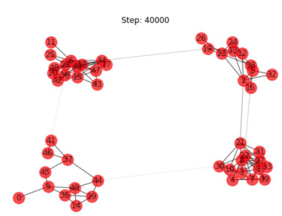
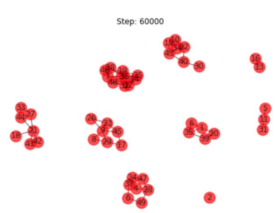
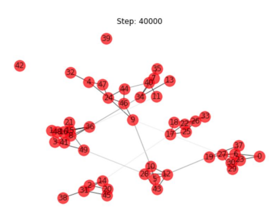
		Watts-Strogatz	Barabasi-Albert
update_1_0	original		
update_1_1	If the weight is < 0.05, decrease the difference of opinions (nodes)		
update_1_2	If the weight is < 0.05, increase the connection/ weights (edges)		
update_2_1	Increase the number of topics/ opinions to 2		
update_2_2	Increase the number of topics/ opinions to 3		

Table 2 - Update Overview

Forcing relationships is more effective than forcing opinions

Table 2 compares the behavior of Both Watts-Strogatz and Barabasi-Albert graph with the different updates. Both show a similar behavior with regards to the formation and topography of nodes. With an initial network of 50 nodes and at 20000 steps, the original update and the opinion update form two clusters, the latter one being more heavily connected though since connections cannot break. Forcing the relationship to bounce to 0.2 when it gets too bad results in the network more quickly homogenizing to similar opinions and forming one cluster. While the comparison has to be tested for more comparative values for epsilon = 10 and weight = 0.2, given these values forcing relationships is more effective at creating a homogenous community than forcing opinions.

More opinions form more clusters

With regards to multiple opinions, both graphs cluster opinions similarly after 40,000 steps. Since opinions can at extreme can take only 2 values, 0 or 1, the number of clusters is determined by the number of topics n.

$$\text{number of clusters} = 2^{\text{number of topics}}$$

Given 2 opinions, there are expected to be 2^2=4 clusters. Given 3 opinions, there are expected to be 2^3=8 clusters. This is clearly visible for two topics. For three topics, around 8-9 clusters can be detected. While this is not as convincing, it is a good approximation of the expected clusters. Increasing the number of nodes or iterations did not change the behavior.

Densities remain equal for Barabasi-Albert graph

After running the simulation with the update functions, Table 3 and 4 show that for both graphs, both update functions increase the densities for the graph. (With density being defined as the proportion of potential connections realized.) However, for the Barabasi-Albert graph the updates result in the same densities (Table 4). This could be due to the preferential attachment, that decreases the difference of the effects of the two update functions. The effects of editing edges and editing opinions, however, are similar in both graphs, and double the effect of the original update function. For example, densitis change from 0.05 to 0.1 for Watts-Strogatz graph, and from 0.1 to 0.2 for Barabasi-Albert graph.

	Update Edges	Update Opinions	Update Original
0	0.099592	0.093878	0.052245
1	0.114286	0.105306	0.055510
2	0.135510	0.115918	0.065306
3	0.150204	0.130612	0.070204
4	0.158367	0.146122	0.077551
5	0.169796	0.163265	0.078367
6	0.196735	0.177959	0.090612
7	0.209796	0.194286	0.093061
8	0.223673	0.209796	0.101224
9	0.238367	0.231020	0.114286

Table 3 - Watts-Strogatz densities

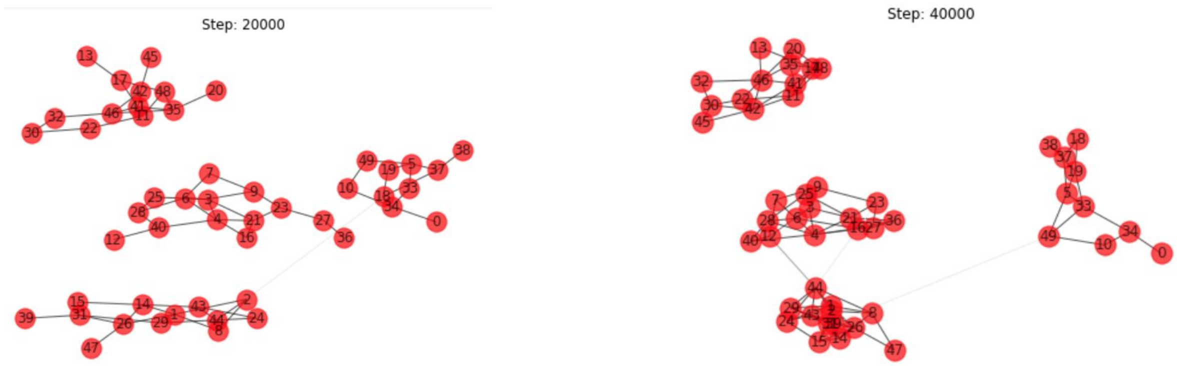
	Update Edges	Update Opinions	Update Original
0	0.200816	0.200816	0.104490
1	0.219592	0.219592	0.114286
2	0.235918	0.235918	0.119184
3	0.253061	0.253061	0.124082
4	0.264490	0.264490	0.127347
5	0.281633	0.281633	0.135510
6	0.293878	0.293878	0.142041
7	0.311837	0.311837	0.149388
8	0.326531	0.326531	0.165714
9	0.345306	0.345306	0.174694

Table 4 - Barabasi-Albert densities

Progression of multiple topics

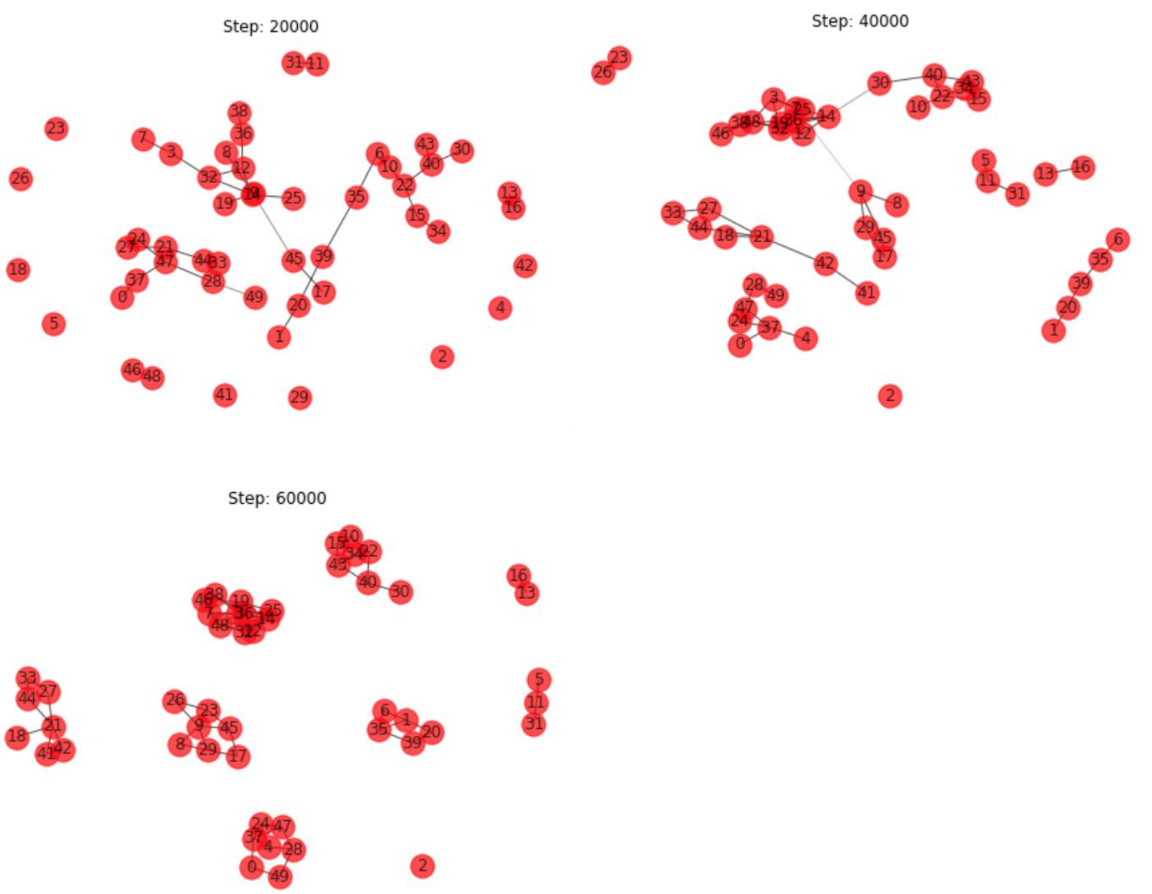
2 topics

Both at 20,000 and 40,000 steps the formation of clusters is visible.



3 topics

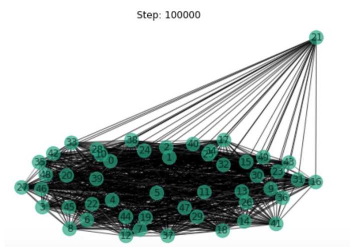
The clusters become more visible as the progression reaches 60,000 steps. This could indicate that the more topics there are, the longer the simulation takes to form clusters. A reason could be that the formation of opinion takes longer.



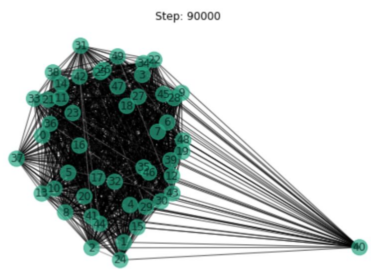
Opinions will always converge

No matter what simulation or update runs, the opinions will always converge, and if connections persist will form one cluster. Since the update functions only allows for the difference of opinions to decrease, but not to increase, the opinions eventually will converge.

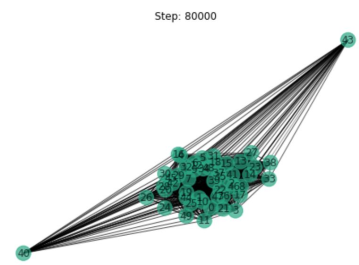
This is a major limitation about the simulation. In real life, opinions very well can distance from each other and become more extreme. However, it would be interesting exploring this behavior, and building a mathematical analysis around The End of History.



Update_0_1



Update_1_1



Update_1_2

Simulation Analysis

To summarize the above observations:

- Preventing splits increases density and clustering.
- The more topics of opinions, the more clusters form (initially).
- Eventually, all opinions will converge.

One interesting aspect that remains unexplored are differing values of epsilon - the opinion convergence magnifier. Higher values of epsilon indicate that alpha becomes higher and thus the opinions converge more rapidly. That is visible in the table, where higher alpha values tend to show higher clusterings, as well as higher densities. For example, at 10,000 steps clustering is 0.59 for an epsilon value of 1, as opposed to 0.70 for an epsilon value of 16. Densities are 0.60 and 0.72, respectively. This indicates that nodes become more closely connected with each other, and have more connections when epsilon is higher. Part of the density increase as steps increase is explained by the fact that no edges can be removed, but random edges continue to be added. However, the increasing value of alpha at the same steps shows the effect of the amplifier.

	Clustering 10000 steps	Clustering 2000 steps	Clustering 4000 steps	Clustering 6000 steps	Clustering 8000 steps
1.000000	0.595365	0.000000	0.222222	0.333333	0.452308
3.142857	0.613445	0.095238	0.222222	0.337662	0.473846
5.285714	0.625397	0.071429	0.218182	0.359684	0.498462
7.428571	0.641141	0.111111	0.227273	0.379447	0.509852
9.571429	0.654655	0.111111	0.217949	0.402174	0.533333
11.714286	0.665165	0.138889	0.238095	0.427536	0.564516
13.857143	0.681682	0.177778	0.275000	0.433333	0.575758
16.000000	0.697013	0.200000	0.301471	0.433846	0.588235

Table 5: Clusterings for different epsilon values for the Barabasi-Albert graph

	Density 10000 steps	Density 2000 steps	Density 4000 steps	Density 6000 steps	Density 8000 steps
1.000000	0.605714	0.102041	0.212245	0.349388	0.469388
3.142857	0.617143	0.115102	0.233469	0.373878	0.483265
5.285714	0.639184	0.124898	0.245714	0.385306	0.503673
7.428571	0.657143	0.137143	0.263673	0.398367	0.521633
9.571429	0.669388	0.155102	0.281633	0.415510	0.539592
11.714286	0.688163	0.167347	0.299592	0.428571	0.555102
13.857143	0.707755	0.173878	0.314286	0.439184	0.572245
16.000000	0.720000	0.194286	0.331429	0.453878	0.592653

Table 6: Densities for different epsilon values for the Barabasi-Albert graph

The above simulations showed that while social phenomena can be approximated (e.g. strengthening ties as opinions converge), it is far from perfect and may contradict the expected (e.g. converging opinions after diverging and clustering). Nonetheless, it is a useful analysis that can help make decisions in fields of policy advising (elections) to marketing analyses (Facebook).